

BAB II

TINJAUAN PUSTAKA DAN FORMAT PENDUKUNG

2.1. Tinjauan Pustaka

Dalam penelitian ini dirancang sebuah alat yang dapat digunakan untuk presensi dengan sistem pengenalan wajah serta menampilkan suhu tubuh orang tersebut. Pada rancangan ini menggunakan kamera untuk pengenalan wajah dan sensor AMG8833 untuk mengukur suhu tubuh yang dikontrol dengan sebuah *Raspberry Pi 4*.

Tujuan peneliti menggunakan *Raspberry Pi 4* ini karena *Raspberry Pi* merupakan kontroler yang kompleks, *Raspberry Pi* memiliki komponen yang hampir serupa dengan komputer pada umumnya. Seperti CPU, GPU, RAM, Port USB, Audio Jack, HDMI, Ethernet, dan GPIO. Untuk tempat penyimpanan data dan sistem operasi *Raspberry Pi* tidak menggunakan *HardDisk Drive* (HDD) melainkan menggunakan Micro SD dengan kapasitas paling tidak 4 GB, sedangkan untuk sumber tenaga berasal dari USB Tipe C dengan sumber daya yang direkomendasikan yaitu sebesar 5V dan minimal arus 700mA. Beberapa penelitian yang berkaitan dengan topik tersebut antara lain dilakukan oleh:

Hasil dari penelitian (Saputra dkk, 2021) menampilkan suhu tubuh dan menentukan tamu atau anggota dari kantor tersebut. Perbedaannya dari komponen adalah kamera dan *thermo gun*, sedangkan pada penelitian ini penulis menggunakan *webcam* dan sensor AMG8833. Kemudian perbedaan dari metode adalah dengan metode *backpropagation* sedangkan penulis menggunakan metode *haar cascade* dan LBPH.

Hasil dari penelitian (Firmansyah dkk, 2021) adalah mengukur suhu tubuh dan presensi menggunakan kartu RFID. Perbedaannya sistem absensi yang digunakan RFID sedangkan pada penelitian ini penulis menggunakan sistem pengenalan wajah, terdapat juga perbedaan pada sensor yang digunakan untuk mengukur suhu, yaitu menggunakan sensor MLX90614 dan sensor AMG8833.

Hasil dari penelitian (Bustomi dan Hariyanto, 2020) adalah mengenali wajah di beberapa keadaan seperti didalam ruangan atau luar ruangan. Perbedaan pada penelitian sebelumnya hanya berfokus pada sistem absensi menggunakan pengenalan wajah, sedangkan pada penelitian penulis selain sistem absensi

menggunakan pengenalan wajah juga ada pengukuran suhu tubuh. Kemudian metode yang digunakan adalah metode LBPH sedangkan pada penelitian penulis menggunakan metode *haar cascade* dan metode LBPH.

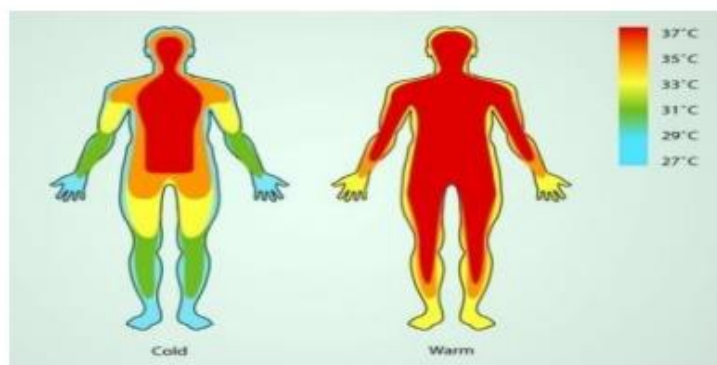
Hasil dari penelitian (Panjaitan dkk, 2021) adalah mengenali wajah dengan menggunakan aksesoris yang dikenakan pada area wajah, dan pengukuran suhu tubuh. pada penelitian sebelumnya kontroler yang digunakan ada 3 yaitu *Raspberry Pi 3*, Arduino, NodeMCU. Sedangkan pada penelitian penulis hanya menggunakan *Raspberry Pi 4* yang sudah bisa mencakupi sistem kerja dari 3 kontroler yang digunakan pada penelitian Panjaitan. Sensor untuk mengukur suhu pada penelitian sebelumnya menggunakan sensor GY-906 sedangkan penulis menggunakan sensor AMG8833.

Hasil dari penelitian (Hartanto dan Prabowo, 2021) adalah sistem absensi yang hanya menggunakan pengukuran suhu, sedangkan penelitian penulis selain menggunakan pengukuran suhu juga menggunakan pengenalan wajah, kontroler yang digunakan, pada penelitian sebelumnya menggunakan Arduino ATmega 2560 sedangkan penelitian penulis menggunakan *Raspberry Pi*. Sensor yang digunakan peneliti sebelumnya untuk mengukur suhu ada sensor MLX90614 sedangkan penulis menggunakan sensor AMG8833.

2.2. Suhu Tubuh Manusia

Suhu tubuh adalah kemampuan tubuh dalam memproduksi dan membuang jumlah panas ke area luar dan dipengaruhi oleh faktor umur, aktivitas, hormon, tingkat stress, dan jenis obat yang telah dikonsumsi. Suhu tubuh manusia terdiri dari suhu inti dan suhu kulit. Suhu tubuh orang dewasa yang normal berkisaran antara 36°C-37,6°C.

Untuk menjaga suhu dalam keadaan stabil, diperlukan regulasi suhu tubuh. Suhu tubuh manusia diatur dengan metode *feedback*. Metode *feedback* ini terjadi bila suhu pada inti tubuh telah melampaui batas dari toleransi tubuh untuk menjaga suhu yang disebut titik tetap.



Gambar 2.1. Suhu Tubuh Manusia
(Sumber: www.mindmedia.com)

2.3. Pendeteksian dan Pengenalan Wajah

Deteksi wajah merupakan sebuah metode dalam mengekstraksi daerah pada wajah dengan tujuan sebagai proses pengenalan wajah. Sementara itu Pengenalan wajah termasuk kedalam sebuah teknologi yang melakukan pengolahan citra (*computer vision*), dalam hal ini dapat mengidentifikasi suatu identitas atau informasi seseorang dari wajahnya. *Face recognition* merupakan sebuah teknologi komputer yang mampu mengukur dan mengidentifikasi wajah melalui sebuah gambar digital dengan melakukan pencocokkan tekstur dari lekuk wajah yang sudah terdata dan tersimpan dalam dataset.

2.4. Pengolahan Citra Digital

Diperolehnya sebuah Citra digital merupakan hasil digitalisasi dari sebuah citra analog. Ada dua proses dalam mendigitalisasi sebuah citra yaitu proses sampling dan kuantisasi. Dalam proses sampling ditunjukkan banyaknya suatu piksel/blok dalam mendefinisikan sebuah gambar. Sedangkan dalam suatu proses kuantisasi akan menunjukkan banyaknya suatu nilai derajat pada setiap piksel (menunjukkan sejumlah bit yang berada pada sebuah gambar digital, dengan 2 bit *black/white*, 8 bit *grayscale*, dan 24 bit *true color*). Pengolahan Citra dilakukan agar dapat melakukan peningkatan kualitas dari citra agar mudah diinterpretasi oleh manusia maupun komputer. Masukannya merupakan berupa sebuah citra dan keluarannya akan berupa citra juga, yang diekstrak menjadi kualitas lebih baik lagi dari pada citra masukkan sebelumnya. Operasi yang berhubungan antara pengolahan citra dalam mengukur wajah adalah *neighborhood operation*, *histogram equalization*,

grayscale, thresholding, resizing. Proses dari suatu analisis sebuah citra wajah dimulai dari melakukan pencitraan sampai dengan proses terakhir, sehingga akan didapatkan sebuah keputusan dengan maksud dan tujuan tertentu. (Bustomi dan Hariyanto, 2020)

2.5. Haar Cascade Classifier

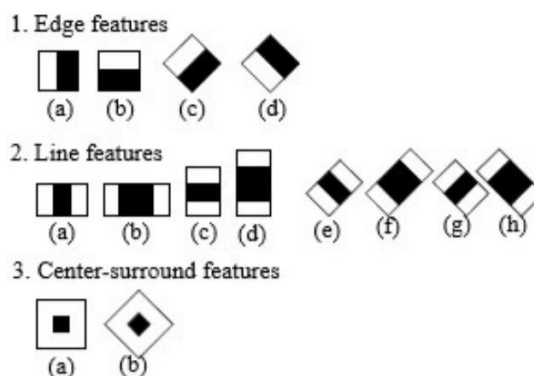
Haar cascade classifier merupakan metode yang digunakan dalam pendeteksian objek. Metode ini juga dikenal dengan metode Viola Jones dikarenakan diperkenalkan oleh Paul Viola dan Michael Jones untuk mendeteksi wajah. Metode ini mendeteksi objek dalam gambar dengan menggabungkan tiga kunci utama yaitu Haar feature, integral image, dan cascade classifier.

2.5.1. Haar Like Feature

Haar like features merupakan rectangular (persegi) *features*, yang memberikan indikasi secara spesifik pada sebuah gambar atau *image*. Ide dari *Haar like features* adalah untuk mengenali objek berdasarkan nilai sederhana dari fitur. Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah *image*. Metode ini merupakan metode yang menggunakan statistika model (classifier). Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat konsep utama yaitu training data, fitur segi empat sederhana yang disebut fitur Haar, integral image untuk pendeteksian fitur secara cepat dan pengklasifikasian bertingkat (*cascade classifier*) untuk menghubungkan banyak fitur secara efisien. Haar like feature digunakan dalam mendeteksi objek pada citra digital. Nama Haar merujuk pada suatu fungsi matematika (Haar Wavelet) yang berbentuk kotak. Awalnya pengolahan gambar hanya dengan melihat dari nilai RGB tiap piksel, namun metode ini ternyata tidaklah efektif. Kemudian dikembangkan sehingga terbentuk Haar-like Feature. (Prathivi dan Kurniawati, 2020)

Haar like feature memproses gambar dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa piksel. Setiap kotak kemudian diproses dan didapatkan perbedaan nilai (*threshold*) yang menandakan daerah gelap dan terang. Nilai-nilai inilah yang nantinya dijadikan dasar dalam *image processing*. Lalu untuk gambar bergerak (video), perhitungan dan penjumlahan piksel terjadi secara terus-menerus

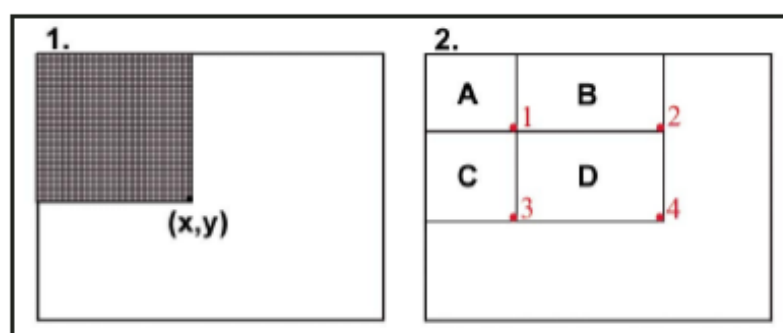
dan membutuhkan waktu yang lebih lama. Oleh karena itu, penjumlahan diganti dengan integral sehingga didapatkan hasil lebih cepat. Hasil deteksi dari *haar like feature* kurang akurat jika hanya menggunakan satu fungsi saja sehingga biasanya digunakan beberapa fungsi sekaligus. Semakin banyak fungsi yang digunakan maka hasilnya akan semakin akurat.



Gambar 2.2. Contoh Variasi Fitur Pada *Haar Like Features*
(sumber: Prathivi dan Kurniawati, 2020)

2.5.2. Integral Image

Integral *image* adalah representasi citra baru yang digunakan untuk untuk menentukan ada atau tidaknya dari fitur haar pada sebuah gambar secara efisien. pada umumnya, pengintegrasian tersebut berarti menambahkan nilai piksel bersamaan. Nilai integral untuk masing-masing piksel adalah jumlah dari semua piksel-piksel dari atas sampai bawah. Dimulai dari kiri atas sampai kanan bawah, nilai dari citra dapat dijumlahkan dengan beberapa operasi bilangan bulat per piksel. (Prathivi dan Kurniawati, 2020)

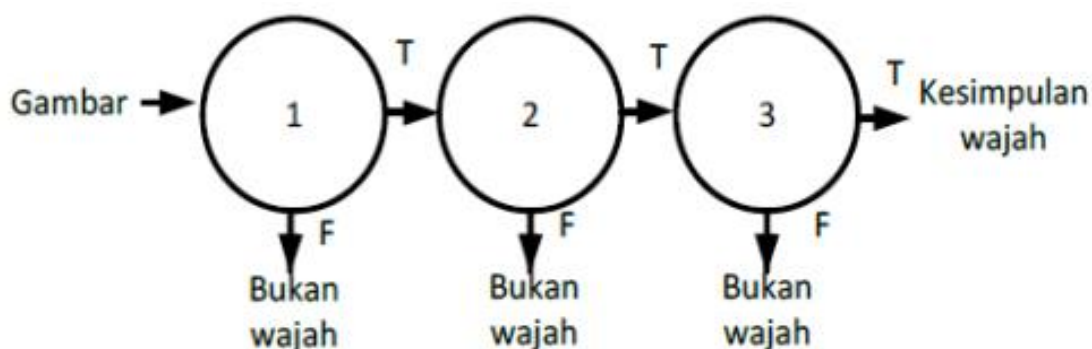


Gambar 2.3. Integral *Image*
(sumber: Prathivi dan Kurniawati, 2020)

Seperti yang ditunjukkan oleh Gambar 2.3 setelah pengintegrasian, nilai pada lokasi piksel (x,y) berisi jumlah dari semua piksel di dalam daerah segiempat dari kiri atas sampai pada lokasi (x,y) atau daerah yang diarsir. Guna mendapatkan nilai rata-rata piksel pada area segiempat (daerah yang diarsir) ini dapat dilakukan hanya dengan membagi nilai pada (x,y) oleh area segiempat.

2.5.3. Cascade Classifier

Cascade classifier adalah metode klasifikasi bertingkat dengan menggunakan fitur yang diseleksi dengan menggunakan semua metode pada *Haar Cascade Classifier*, cascade classifier memiliki beberapa tingkatan dalam melakukan klasifikasi, pada setiap tingkatan dipisahkan antara subcitra yang mengandung gambar positif (gambar yang memiliki objek yang diinginkan) dengan gambar negatif (gambar yang tidak memiliki objek yang diinginkan), dimana bagian subcitra yang tidak mengandung objek positif akan ditolak sedangkan subcitra yang mengandung objek akan dijadikan input pada tingkatan klasifikasi berikutnya dengan kriteria penyaringan yang lebih spesifik hingga didapatkan subcitra yang merupakan objek yang terdeteksi.



Gambar 2.4. Alur kerja metode *Haar Cascade Classifier*
(Sumber: Ramadini dan Haryatmi, 2022)

Cascade Classifier merupakan step untuk mendapatkan hasil yang lebih akurat dengan menghitung nilai *Haar Feature* dalam jumlah banyak dan berulang. Gambar 2.1 menampilkan alur kerja dari *Cascade Classifier*.

Haar Cascade dapat diprogram untuk mendeteksi beberapa objek, yang harus dilakukan adalah dengan menentukan area pada wajah yang memiliki kemungkinan tertinggi. Wajah tersebut memiliki kulit dan memiliki tingkat piksel warna pada kulit. Pemilihan teknik segmentasi dipilih untuk warna piksel pada

wajah. Kemudian memvalidasinya dengan *haar cascade classifier*, Jika piksel yang divalidasinya sesuai dengan geometriunya maka sistem telah menemukan wajah yang dimaksud, jika tidak sesuai maka sistem mengabaikannya

2.6. Local Binary Pattern Histogram (LBPH)

Pengenalan wajah merupakan proses lanjutan dari proses pendeteksian wajah. Di dalam pendeteksian wajah yaitu deteksi bagian wajah dari seseorang, wajah tersebut bisa didapatkan dari gambar maupun video. Dengan memanfaatkan hasil *training* dari *haar cascade*. Kemudian hasil dari proses ini dikombinasikan dengan proses *Image Matching* dengan algoritma LBPH. Dengan metode ini, foto yang sudah di-learning akan dicocokkan dengan hasil mengukur dari *streaming* kamera dimana pada *streaming* nantinya beberapa gambar dalam dataset kemudian dicocokkan dengan memanfaatkan nilai histogram yang telah diekstraksi dari gambar dengan memanfaatkan persamaan LBPH. (Ramadini dan Haryatmi, 2022)

Citra wajah yang diambil secara *realtime* menggunakan kamera pada laptop akan dibandingkan dan dicocokkan menggunakan histogram yang sudah diekstraksi dengan citra wajah yang ada pada database. LBPH merupakan sebuah fitur yang berfungsi untuk mengklasifikasi citra yang dikombinasikan dengan histogram dan LBPH salah satu teknik terbaru dari metode LBP yang digunakan untuk mengubah performa hasil pengenalan wajah. Cara kerja LBPH ditunjukkan pada Gambar 2.5 Yang menjelaskan bahwa piksel yang berada di tengah diperoleh dengan cara membandingkan intensitas piksel yang lain. Nilai dari piksel yang berada ditengah adalah ambang batas dari kedelapan piksel yang lainnya.

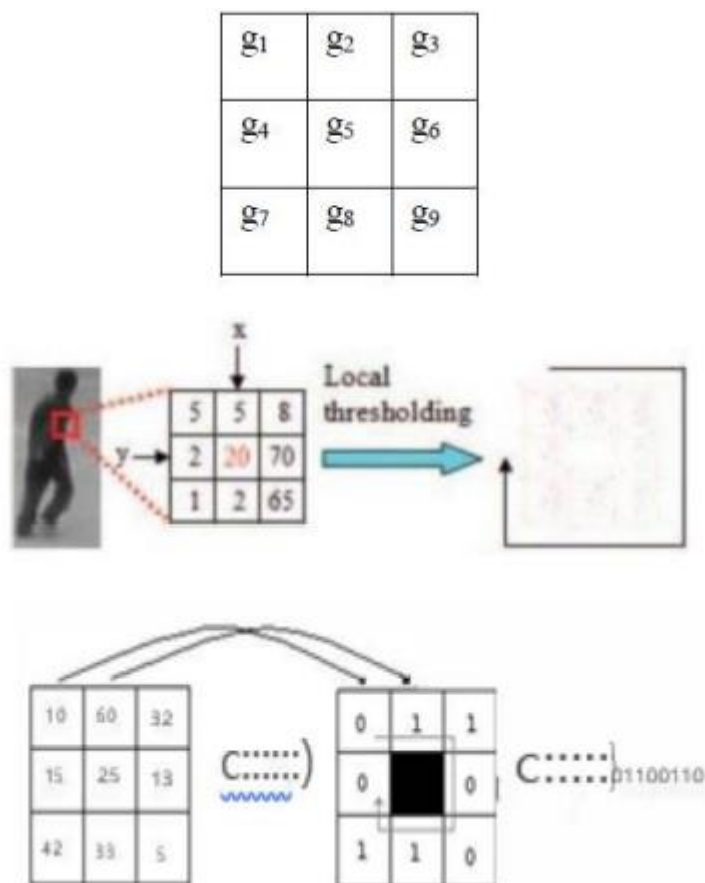
LBPH merupakan kombinasi dari metode *Local Binary Pattern* (LBP) dengan *Histograms of Oriented Gradients* (HOG). LBP adalah operator tekstur yang sederhana akan tetapi sangat efisien yang memberi label pixel suatu gambar dengan cara membedakan lingkungan dari setiap pixel dan menganggap hasilnya sebagai angka biner sedangkan HOG akan meningkatkan kinerja pendeteksian berdasarkan histogram.

Pada metode LBP terdapat 6 langkah yaitu:

1. Ubah citra wajah menjadi citra grayscale dengan range intensitas piksel adalah 0-255.
2. Membagi citra wajah menjadi bagian-bagian kecil yang disebut dengan sel, dimana tiap sel berukuran 3x3 piksel. Karena tiap sel berukuran 3x3 maka banyaknya tetangga (P) adalah 8 dan radius (R) dari piksel tengah adalah 1.
3. Tentukan nilai pusat dari tiap sel yaitu piksel tengah untuk digunakan sebagai nilai ambang. Nilai tersebut digunakan untuk menentukan nilai-nilai baru dari ke-8 pixel disekitarnya (piksel tetangga).
4. Untuk setiap piksel tetangga dari piksel tengah, akan ditetapkan nilai biner baru dengan aturan 1 untuk nilai yang sama atau lebih tinggi dari ambang dan 0 untuk nilai lebih rendah dari ambang.
5. Nilai piksel pada citra wajah selain piksel tengah hanya akan berisi nilai-nilai biner. Nilai biner tersebut digabungkan dari setiap posisi menjadi barisan biner baru yang berukuran 8 bit (contoh: 10001101)
6. Kemudian, nilai biner ini akan dikonversikan ke nilai desimal dengan range 0-255 sehingga menghasilkan citra baru yang mewakili karakteristik (fitur) dari citra asli dengan lebih baik. Hasil operator LBP diperoleh dari persamaan.

Setelah proses LBP dilakukan, selanjutnya dilakukan ekstraksi histogram dengan menggunakan metode Histograms of Oriented Gradients (HOG). Pada metode HOG terdapat 4 langkah yaitu:

1. Citra hasil LBP dibagi menjadi daerah-daerah kecil yang disebut dengan grid.
2. Buat histogram dari tiap daerah tersebut.
3. Karena citra hasil LBP merupakan citra grayscale maka setiap histogram dari setiap grid hanya akan berisi 256 posisi (0-255) yang mewakili setiap intensitas piksel.
4. Histogram dari tiap daerah (grid) digabung menjadi satu histogram besar. Jika tiap grid berukuran 8x8, maka total memiliki $8 \times 8 \times 256 = 16,384$ posisi dalam histogram akhir. Histogram terakhir merupakan fitur yang mewakili citra asli dan diasumsikan sebagai sebuah vektor.



Gambar 2.5. Cara Kerja Metode LBPH
(Sumber: Ramadini dan Haryatmi, 2022)

Pada gambar cara kerja LBPH dijelaskan bahwa piksel yang berada di tengah diperoleh dengan cara membandingkan intensitas citra yang ditangkap kamera dengan intensitas piksel yang lain. Nilai dari piksel yang berada di tengah adalah nilai ambang batas dari kedelapan piksel yang lainnya. Pada sebuah matriks tersebut nilai biner yang berada di tengah akan dibandingkan dengan nilai yang ada disekelilingnya. Jika nilai pada matriks yang berada di tengah lebih tinggi dari nilai yang ada disekelilingnya, maka nilai matriks di sekelilingnya akan bernilai '1' begitupun sebaliknya jika nilai pada matriks yang berada di tengah lebih rendah dari nilai yang ada disekelilingnya, maka nilai matriks di sekelilingnya akan bernilai '0'. Kemudian dihitung nilai histogram untuk membandingkan dan mencocokkan wajah yang ada pada kamera dengan yang ada di database.

2.7. Python

Python adalah salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung (interpretatif) dengan metode orientasi objek (*Object Oriented Programming*) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan syntax. Sebagian lain mengartikan Python sebagai bahasa yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Walaupun Python tergolong bahasa pemrograman dengan level tinggi, nyatanya Python dirancang sedemikian rupa agar mudah dipelajari dan dipahami.

Python sendiri menampilkan fitur-fitur menarik sehingga layak untuk dipelajari. Pertama, Python memiliki tata bahasa dan script yang sangat mudah untuk dipelajari. Python juga memiliki sistem pengelolaan data dan memori otomatis. Selain itu modul pada Python selalu di *update*. Ditambah lagi, Python juga memiliki banyak fasilitas pendukung. Python banyak diaplikasikan pada berbagai sistem operasi seperti Linux, Microsoft Windows, Mac OS, Android, Symbian OS, Amiga, Palm dan lain-lain.

Kode-kode yang ada didalamnya mudah dibaca dan dapat menjalankan banyak fungsi kompleks dengan mudah karena banyaknya *standard library*. Pengembangan program pada Python pun dapat dilakukan dengan cepat dan menggunakan lebih sedikit kode. Bahkan Python mampu menjadikan program dengan skala sangat rumit menjadi mudah. Python sendiri mendukung *multiplatform* dan *multi system* serta memiliki sistem pengelolaan memori otomatis seperti Java.

2.8. Visual Studio Code

Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java, dst). Banyak sekali fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya Intellisense, Git

Integration, Debugging, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VSCode dengan teks editor-teks editor yang lain. Teks editor VSCode juga bersifat *open source*, yang mana kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Kode sumber dari VSCode ini pun dapat dilihat di link Github. Hal ini juga yang membuat VSCode menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan VSCode ke depannya.

2.9. Virtual Network Computing (VNC)

Virtual Network Computing merupakan sebuah *software* remote kontrol, dimana dengan melalui software ini suatu komputer dapat melakukan akses untuk bekerja di suatu komputer lain yang terhubung dengan jaringan. Pada grafis desktop sharing sistem yang menggunakan remote Bingkai Buffer protocol (RFB) untuk kontrol jarak jauh lain komputer. Ini mentransmisikan Keyboard dan Mouse dari satu komputer ke komputer lain, merelay grafis layar pembaruan kembali ke arah lain, melalui jaringan. Perangkat lunak ini terdiri dari sebuah server dan aplikasi klien untuk mengontrol layar komputer lain dari jarak jauh. (Ariyanto, 2015)

Software VNC ini dapat digunakan dengan baik di lingkungan Windows, Linux, Beos, Macintosh, Unix dll. bahkan penggunaannya juga dapat dilakukan secara lintas platform. VNC client & VNC server dapat saling diakses misalnya dari sistem Windows ke sistem Linux, maupun dari sistem Linux ke sistem Windows.